Self-Tuning Database Systems

ISSDM Renewal Grant Proposal

PI: Neoklis Polyzotis <alkis@cs.ucsc.edu>

1 Project Summary

It is common knowledge that the modern world generates and stores information at an increasing rate. A typical example is the emerging field of e-science, where the adoption of high-performance computing has led to the generation of vast volumes of scientific data. The usefulness of the stored information, of course, depends crucially on the ability to query it effectively, and hence on the use of state-of-the-art data management technology.

Unfortunately, the adoption of database systems in e-science has been less than spectacular and has often lead to a lot of frustration among the intended users. One key reason is the complexity of managing and tuning a database system. Essentially, the performance of a database system depends crucially on its physical design, i.e., the set of physical structures such as indices and materialized views, that can speed up the execution of queries. An effective physical design has to match the traits of the workload, e.g., it must index more heavily the parts of the database that are frequently referenced in queries, and at the same time it must fit within an alloted disk space budget. Thus, designing an effective physical design involves a non-trivial optimization problem: maximize the throughput of the query processor, assuming limited resources for storing the materialized structures. This is a challenging task even for an experienced database administrator, let alone for a scientist (or, for that matter, any non-expert user) who is not likely to have the necessary expertise.

To address this important issue, we propose to develop a database system that can self-organize its physical configuration without the intervention of a human administrator. The key idea is to augment the system with an on-line tuning module that monitors continuously the incoming workload, gathers statistics on the performance of the system, and reorganizes the physical design periodically in order to maximize query throughput. Clearly, this is a challenging goal. Since tuning happens concurrently with normal database operation, the tuner has to operate with low and controllable overhead so that it does not affect query performance. Moreover, the tuning module has to track continuously the traits of the workload and identify potential physical designs that can improve performance. Even more importantly, the tuner has to determine whether a recent change in the workload constitutes a major shift in the query distribution that justifies the potentially high cost of changing the physical design. These problems stem from the on-line nature of the problem and have not been addressed by previous studies on database tuning. Thus, the proposed research will make significant scientific contributions in the area of self-tuning systems and autonomic computing.

The proposed project forms part of our more general agenda on the development of database technology that targets specifically non-expert users, and in particular scientists. In brief, our goal is to lower the bar for the adoption of database technology in the field of scientific data management. The proposed research is thus relevant to projects in LANL that involve the management and querying of large volumes of experimental and scientific data.

2 Project Description

Overview of Existing Technology for Physical Design Tuning

Earlier studies [5, 7, 28] have introduced techniques that recommend a physical design by analyzing a representative query workload. This paradigm is typically referred to as *off-line tuning*, since the workload is gathered and analyzed before the database system goes live. The use of a representative workload implies that off-line tuning is suitable for applications where the query load has certain predominant characteristics. Typical workloads over scientific data exhibit the inverse trait, since queries are often generated by human users in the course of a data analysis task.

Acknowledging the shortcomings of off-line tuning, previous studies have advocated an on-line approach to physical design tuning. The basic idea is that tuning is performed continuously in order to keep track of the current traits of a volatile query workload. Earlier studies have developed rudimentary on-line tools for index selection in relational databases [9, 8], whereas more recent studies [19, 6] have introduced more sophisticated solutions. These recent techniques provide interesting insights into the problem of selecting indices on-line, but they focus on specific variants of the problem and thus lack generality. Moreover, they do not address the issue of regulating the overhead of tuning, a point that is clearly important for the integration of on-line tuning in real-world data management applications.

Completed Work

COLT As part of our ongoing work on this project, we have developed the COLT [22] framework (short for *Continuous On-Line Tuning*) that supports the on-line selection of index structures. COLT is realized as a separate system module that builds a model of the current workload (based on the incoming query flow), estimates the respective gains of different candidate indices, and selects those that would provide the best performance for the observed workload within the available storage. Thus, the system performs continuous profiling and reorganization in order to materialize indices that match the most recent traits of the query workload. A key novelty of COLT is that it incorporates mechanisms to regulate the overhead of on-line tuning based on its potential to improve performance. By using these mechanisms, COLT lowers its overhead if the workload is stable and the system is well-tuned, and intensifies tuning when a shift is detected and the system has to adapt to a new workload. We have completed a prototype implementation of COLT inside the PostgreSQL open-source database system, and we have performed an experimental study on the effectiveness of our technique. Overall, our results have shown that COLT can adapt the index set to the current traits of the workload, while incurring a low overhead on normal query execution. We do not discuss further the details of our technique in the interest of space. A more complete description can be found in the respective publications [20, 21, 22].

A Benchmark for Online Index Selection As mentioned earlier, several recent studies (including our own work) have advocated an online approach to the problem of index selection. The proposed tuning algorithms [6, 19, 22] cover a wide gamut of features with respect to the type of indices that can be materialized, the performance guarantees enabled by the algorithm, and the complexity of the workloads that can be handled. However, each algorithm has been evaluated in isolation in the respective study, using a different methodology and execution environment. This makes it difficult to draw conclusions on the relative merits of the proposed techniques. Motivated by this observation, we developed a principled benchmarking methodology for evaluating the performance of online tuning algorithms in a controlled execution environment. The proposed benchmark consists of several workload suites, designed to exercise specific aspects of a tuning algorithm. The workloads are constructed using a general methodology that can be used to generate additional interesting suites. Using the benchmark, we conducted an empirical evaluation using the PostgreSQL DBMS of two representative online tuning algorithms, namely the algorithm of Bruno and Chaudhuri [6] and COLT [21]. The results revealed interesting insights on the relative merits of the two algorithms and their performance characteristics under different scenarios. It is also interesting to note that COLT outperformed the other algorithm in most experiments, even though COLT is less powerful "on paper". We view this as evidence in favor of less complicated approaches to online tuning.

Modeling and Analysis of Index Interactions. Index interactions are a complicating factor in the problem of index design tuning. In a nutshell, two indexes interact if their joint benefit is different than the sum of the individual benefits. For instance, two indexes may have a low benefit if used in isolation, but a high benefit if used together in an index intersection. Modeling and analyzing index interactions is crucial for effective index tuning, since it enables the tuning algorithm to track more accurately the joint benefit of an index set. Knowledge about index interactions can also benefit a database administrator who is interested in understanding the performance characteristics of a specific index design. Motivated by these observations, we developed a formal framework for modeling and analyzing benefit interactions in the context of index tuning [25]. The proposed framework is the first systematic solution for computing benefit interactions among the indexes in a recommended index design. The study also proposes two novel database administration tools. The first is an intuitive method to visualize index interactions. Using the visualization method, the administrator can understand better the performance characteristics of the recommended indexes and can thus make more informed decisions on the final design. The second tool computes a schedule to construct a specific index set. The tool takes into account index interactions in order to maximize the accumulated benefit after each materialization.

Proposed Research

The proposed research aims to develop robust self-tuning algorithms for the problem of index selection. In this direction, we will leverage the results from our previous work. In what follows, we outline three initial directions towards our goal.

Online Tuning using the Work Function Algorithm. The problem of online index selection bares similarities to the well studied problem of *task systems* from the field of online optimization. A task system can be loosely described as follows. Consider a sequence of computational tasks $\bar{T} = T_1, \ldots, T_n$ that is given to a system for processing. We assume that the system executes one task at a time and can observe task T_{i+1} only after it completes the execution of T_i. We associate with the system a set $\mathcal{S}$ of possible states, and we use $S_i \in \mathcal{S}$ to denote the state of the system when T_i is processed. Now, the cost of processing task T_i depends on the choice of S_i and is denoted as $\mathit{taskCost}(T_i, S_i)$. Moreover, there is a cost $\mathit{transCost}(S_{i-1}, S_i)$ for transitioning to a state S_i from the previous state S_{i-1}. Given a schedule of states $\bar{S} = S_1, \ldots, S_n$ that specifies the state of the system for each task, we use $\mathit{totalCost}(\bar{S}, \bar{T})$ to denote the total processing cost (task processing + state transitioning).

Clearly, the description of a task system matches well the problem of physical design tuning. The query processor (system) receives a sequence of queries (tasks), and the execution cost of each query depends on the currently installed physical design (state). The goal is to determine a schedule of physical designs in order to minimize the overall cost of processing queries and transitioning between designs. This scheduling problem has been investigated extensively in the context of task systems and several interesting algorithms have been developed that could form the basis of an online tuner. One particularly attractive algorithm is the well known *Work Function Algorithm* [4] (WFA, for short), which has been shown to perform within a factor of $2|\mathcal{S}| - 1$ of the optimal clairvoyant algorithm, i.e., an ideal algorithm that has complete knowledge of the workload. This part of our project aims at developing a tuning framework using the robust WFA as the foundation.

One important obstacle in the application of WFA is that it assumes a symmetrical transition cost function. This is clearly false in the physical design problem, since the cost to drop physical structures that belong in the difference of two states is zero, but the cost to materialize them is positive. As a first step in our work, we have shown that the characteristics of the online tuning problem render this assumption unimportant, and thus it is possible to maintain the competitive ratio of $2|\mathcal{S}| - 1$ with an asymmetric transition cost function. A second challenge is that the computational overhead of WFA grows linearly with the size of $\mathcal{S}$, which is quite large for the problem of physical design tuning. In the problem of index selection, for instance, $\mathcal{S}$ is the power-set of the set of indices that are relevant to the workload. In this direction, we have been able to identify conditions under which WFA can be run efficiently without tracking every possible subset of indices. It is interesting to note that our solution leverages information on index interaction, and hence the ideas outlined in previous paragraphs are directly applicable here. As part of our proposed research, we will finalize the details of the tuning algorithm and evaluate its performance using the benchmark suite that we have developed.

Online Tuning in Parallel Systems. Up to this point, our formulation of the index selection problem has targeted a centralized scenario of a single query processor. In this part of the proposed research plan, we will investigate the application of online tuning in a parallel system of shared-nothing query processors. A shared-nothing system partitions data across several nodes and evaluates a query by running a parallel sub-query locally on each participating node. This system architecture offers several benefits compared to other alternatives (e.g., shared-memory or shared-disk) and is thus frequently encountered in practice.

Following up on our current work, we will again focus on the problem of online index selection. One option is to perform *localized* tuning in order to install indices on each node that speed-up the evaluation of the sub-queries received by the node. This can be achieved by running an independent copy of our tuning algorithm on each node. A more promising and interesting alternative is to tune the set of indices at a *global* level, by taking advantage of data replication. More concretely, shared nothing systems replicate data aggressively in order to ensure availability. (In fact, one may argue that replication is necessary in this context given the high rate of node failure in large clusters.) The existence of replicas raises the opportunity of physical design specialization: Each node that holds replicated data may maintain a different set of indices in order to specialize for a subset of the workload. As an example, assume that relation R is replicated on nodes A and B. Node A can maintain an index on $R.a$ in order to efficiently answer queries on this specific attribute, while node B can create indices tailored to a different subset of queries. This approach takes advantage of the aggregated disk storage in the two nodes in order to improve query performance. In contrast, if queries are routed in a simple round-robin fashion between the two nodes, then localized tuning would install the same set of indices which would benefit a smaller subset of the global workload.

The challenge in solving this tuning problem is that the specialization of each node determines the part of the workload that it can benefit and vice versa. As a first step in solving this interesting problem, we plan to investigate the offline approach where a representative workload is known in advance. Subsequently, we will expand our techniques to the online version of the problem, where the workload is volatile and the specialization has to be modified on-the-fly. We expect that our work on index interaction will prove useful in the development of smart tuning algorithms.

Online Index Tuning for Visualization Pipelines. In collaboration with Jon Goodring from LANL, we propose to investigate the application of our index tuning techniques on the optimization of visualization pipelines.

Scientists rely heavily on visualization tools to explore and analyze experimental data. The visualizations are typically generated by a processing pipeline that takes as input the original data, applies a series of filters that restrict and transform the data, and generates as output the final visual. The pipeline is executed on demand, based on the actions of the scientist on the visualization tool. It is common for the pipeline to be executed several times in the same session, as the user requests to examine different portions or aspects of the data set.

An obvious goal of any visualization tool is to maintain an interactive user experience. However, the cost of executing the visualization pipeline can be significant, particularly for the large data sets encountered in scientific data management. As part of this project, we will investigate optimizations that can reduce drastically the execution cost of a visualization pipeline. One source of inefficiency in existing visualization systems is that the pipeline consumes the complete data set, i.e., a full scan of the data is performed, even though the pipeline filters may select only a subset of the data. We will thus develop suitable data indexing techniques so that the pipeline accesses only the portion of the data that is relevant to its output. The indexes will be created in an online fashion, by taking into account the interaction of the user with the visualization interface and how this maps to filters for the visualization pipelines. Hence, the idea is to index the data where it "matters" according to the needs of the user. This part of the project will leverage our previous research results in online index tuning and also in speculative query processing for relational databases [13].

3 Personnel

The PI of the project is Neoklis Polyzotis, an associate professor at UC Santa Cruz. The PI's research has focused on self-tuning database systems [24, 1, 20, 2, 22, 13, 3], approximate query answering for semi-structured and relational data [10, 23, 16, 27, 26, 17, 11, 12, 18, 15], and active data warehousing [14].

The project will fund a graduate student researcher. The student will be mentored jointly by the PI and the following researchers from LANL: John Bent, Gary Grider, and Jon Woodring.

References

[1] S. Abiteboul, Ioana Manolescu, Neoklis Polyzotis, Nicoleta Preda, and Chong Sun. Xml processing in dht networks. In *Proceedings of IEEE ICDE 2008*, pages 606–615, 2008.

[2] S. Abiteboul and N. Polyzotis. Data ring: Let us turn the net into a database! In Yannis Manolopoulos, Jaroslav Pokorný, and Timos K. Sellis, editors, *ADBIS*, volume 4152 of *Lecture Notes in Computer Science*, page 1. Springer, 2006.

[3] S. Abiteboul and N. Polyzotis. The Data Ring: Community Content Sharing. In *Conference on Innovative Data Systems Research (CIDR)*, 2007.

[4] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 2005.

[5] N. Bruno and S. Chaudhuri. Automatic Physical Database Tuning: A Relaxation-based Approach. 2005.

[6] Nicolas Bruno and Surajit Chaudhuri. An online approach to physical design tuning. In *ICDE '07: Proceedings of the 23rd International Conference on Data Engineering*, 2007.

[7] S. Chaudhuri and V. R. Narasayya. An Efficient Cost-Driven Index Selection Tool for Microsoft SQL Server. pages 146–155, 1997.

[8] Martin R. Frank, Edward Omiecinski, and Shamkant B. Navathe. Adaptive and automated index selection in rdbms. In *EDBT '92: Proceedings of the 3rd International Conference on Extending Database Technology*, pages 277–292, London, UK, 1992. Springer-Verlag.

[9] Michael Hammer and Arvola Chan. Index selection in a self-adaptive data base management system. In *ACM SIGMOD International Conference on Management of Data*, pages 1–8, 1976.

[10] Veronica Mayorga and Neoklis Polyzotis. Sketch-based summarization of ordered xml streams. In *ICDE*, 2009.

[11] N. Polyzotis and M. Garofalakis. Statistical Synopses for Graph Structured XML Databases. In *Proceedings of the 2002 ACM SIGMOD Intl. Conf. on Management of Data*, pages 358–369, 2002.

[12] N. Polyzotis and M. Garofalakis. Structure and Value Synopses for XML Data Graphs. In *Proceedings of the 28th Intl. Conf. on Very Large Data Bases*, pages 466–477, 2002.

[13] N. Polyzotis and Y. Ioannidis. Speculative query processing. In *Proceedings of the First Biennial Conference on Innovative Data Systems Research*, 2003.

[14] N. Polyzotis, Spiros Skiadopoulos, Panos Vassiliadis, Alkis Simitsis, and Nils-Erik Frantzell. Supporting streaming updates in an active data warehouse. In *Proceedings of IEEE ICDE 2007*, April 2007.

[15] Neoklis Polyzotis and Minos Garofalakis. Xsketch synopses for xml data graphs. *ACM Trans. Database Syst.*, 31(3):1014–1063, 2006.

[16] Neoklis Polyzotis, Minos Garofalakis, and Yannis Ioannidis. Approximate xml query answers. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 263–274, 2004.

[17] Neoklis Polyzotis, Minos Garofalakis, and Yannis Ioannidis. Selectivity Estimation for XML Twigs. In *Proceedings of the 20th Intl. Conf. on Data Engineering*, pages 264–275, 2004.

[18] Neoklis Polyzotis and Minos N. Garofalakis. XCluster Synopses for Structured XML Content. page 63. IEEE Computer Society, 2006.

[19] Kai-Uwe Sattler, Martin Lühring, Ingolf Geist, and Eike Schallehn. Autonomous management of soft indexes. In *SMDB '07: Proceedings of the 2nd International Workshop on Self-Managing Database Systems*, 2007.

[20] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis. Colt: continuous on-line tuning. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 793–795, 2006.

[21] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis. On-line Database Tuning. Technical Report UCSC-CRL-06-07, UC Santa Cruz, 2006.

[22] K. Schnaitter, S. Abiteboul, T. Milo, and Neoklis Polyzotis. On-line index selection for shifting workloads. In *2nd International Workshop on Self-Managing Database Systems (SMDB)*, 2007.

[23] K. Schnaitter, Joshua Spiegel, and Neoklis Polyzotis. Depth estimation for ranking query optimization. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB)*, pages 902–913, September 2007.

[24] Karl Schnaitter and Neoklis Polyzotis. A Benchmark for Online Index Selection. In *4th International Workshop on Self-Managing Database Systems*, 2009.

[25] Karl Schnaitter, Neoklis Polyzotis, and Lise Getoor. Index interactions in physical design tuning: Modeling, analysis, and applications. *PVLDB*, 2(1):1234–1245, 2009.

[26] Joshua Spiegel and Neoklis Polyzotis. Graph-based synopses for relational selectivity estimation. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 205–216, 2006.

[27] Joshua Spiegel, Emmanuel Pontikakis, Suratna Budalakoti, and Neoklis Polyzotis. Aqax: a system for approximate xml query answers. In *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*, pages 1159–1162. VLDB Endowment, 2006.

[28] D. C. Zilio, J. Rao, S. Lightstone, G.M. Lohman, A. Storm, C. Garcia-Arellano, and S. Fadden. DB2 Design Advisor: Integrated Automatic Physical Database Design. pages 1087–1097, 2004.

Biographical Sketch: Neoklis Polyzotis

Professional Preparation

National Technical Univ. of Athens	Electrical and Computer Engineering	Diploma (1997)
Univ. of Wisconsin at Madison	Computer Sciences	M.Sc. (2000)
Univ. of Wisconsin at Madison	Computer Sciences	Ph.D. (2003)

Appointments

Associate Professor, Computer Science Department, University of California at Santa Cruz	2009–present.
Assistant Professor, Computer Science Department, University of California at Santa Cruz	2003–2009.

Selected publications related to proposal

- S. Brandt, C. Maltzahn, N. Polyzotis, and W.C. Tan, "Fusing Data Management Services with File Systems". In *Proceedings of the 4th Petascale Data Storage Workshop*, Supercomputing 2009.
- K. Schnaitter, N. Polyzotis, and L. Getoor, "Index Interactions in Physical Design Tuning: Modeling, Analysis, and Applications". In *Proceedings of the VLDB Endowment (PVLDB)*, 2(1), 2009.
- K. Schnaitter and N. Polyzotis, "A Benchmark for Online Index Selection". In *Proceedings of the 4th International Workshop on Self-Managing Database Systems*, 2009.
- S. Abiteboul, I. Manolescu, N. Polyzotis, N. Preda, and C. Sun. "XML Processing in DHT Networks". In *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, pages 606-615, 2008.
- K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis. "On-line index selection for shifting workloads". In *2nd International Workshop on Self-Managing Database Systems (SMDB)*, 2007.
- S. Abiteboul and N. Polyzotis. "The Data Ring: Community Content Sharing". In *Proceedings of the 3rd Biennial Conference on Innovative Data Systems Research (CIDR)*, 2007. (To appear.)
- S. Abiteboul and N. Polyzotis. "Data ring: Let us turn the net into a database!". In *Advances in Databases and Information Systems, 10th East European Conference, ADBIS 2006*, 2006. (Proceedings published as volume 4152 of Lecture Notes in Computer Science. Springer, 2006.)
- K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis. "COLT: Continuous On-Line Database Tuning". In *Proceedings of ACM SIGMOD 2006* (Demonstration Track), pages 793–795, 2006.

Other recent publications

- O. Greenshpan, T. Milo, and N. Polyzotis, "Autocompletion for Mashups". In *Proceedings of the VLDB Endowment (PVLDB)*, 2(1), 2009.
- G. Candea, N. Polyzotis, and R. Vingralek, "A Scalable, Predictable Join Operator for Highly Concurrent Data Warehouses", In *Proceedings of the VLDB Endowment (PVLDB)*, 2(1), 2009.
- J. Finger and N. Polyzotis, "Robust and Efficient Algorithms for Rank Join Evaluation", In *Proceedings of ACM SIGMOD International Conf. on Management of Data*, 2009.
- V. Mayorga and N. Polyzotis. "Sketch-based Summarization of Ordered XML Streams". To appear in *Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE)*, 2009.
- K. Schnaitter and N. Polyzotis. "Evaluating Rank Joins with Optimal Cost". In *Proceedings of the 27th symposium on Principles of database systems*, pages 43–52,2008.

- K. Schnaitter, J. Spiegel, and N. Polyzotis. "Depth estimation for ranking query optimization". In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB)*, pages 902–913,2007.
- N. Polyzotis, S. Skiadopoulos, P. Vassiliadis, A. Simitsis, and N.E. Frantzell. "Supporting Streaming Updates in an Active Data Warehouse". In *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, pages 476–485,2007.
- N. Polyzotis and M.N. Garofalakis. "XCluster Synopses for Structured XML Content". In *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE)*, pages 63–74, 2006.
- J. Spiegel and N. Polyzotis. "Graph-based synopses for relational selectivity estimation". In *SIGMOD'06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 205–216, 2006.
- N. Polyzotis and M. Garofalakis. "XSketch Synopses for XML Data Graphs". In *ACM Trans. Database Syst.*, 31(3):1014–1063, 2006.

Professional Service

- Awards: NSF CAREER Award, IBM Faculty Development award (2005 and 2006).
- Program Committee Co-chair:
  ACM International Workshop on Web Information and Data Management (2007, 2008)
  VLDB PhD Workshop (2008)
  VLDB Demonstration Track (2009)
- Proceedings chair: ACM SIGMOD International Conference on Management of Data (2006).
- Program Committee Member:
  WWW (2010)
  ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2007)
  VLDB Conference on Very Large Databases (2005,2007)
  ACM SIGMOD International Conference on Management of Data (2005, 2009)
  IEEE International Conference on Data Engineering – ICDE (2005,2008,2009,2010)
  IEEE International Conference on Data Mining – ICDM (2004,2005)
  International Conference on Extending Database Technology – EDBT (2006,2010)
  ACM Conference on Information and Knowledge Management – CIKM (2006)

Collaborators and Co-Editors

Serge Abiteboul, INRIA Futurs, France.
Deepavali Bhagwat, Department of Computer Science, University of California Santa Cruz.
Scott Brandt, Univ. of California Santa Cruz.
George Candea, EPFL, Switzerland.
Surajit Chaudhuri, Microsoft Research.
Jonathan Finger, University of California Santa Cruz.
Minos Garofalakis, Intel Research Berkeley.
Lise Getoor, Univ. of Maryland at College Park.
Vagelis Hristidis, School of Computing and Information Sciences, Florida International University.
Carlos Maltzahn, Univ. of California Santa Cruz.
Ioana Manolescu, INRIA Futurs, France.
Veronica Mayorga, University of California Santa Cruz.

Yannis Ioannidis, Department of Informatics, University of Athens, Greece.
Yossi Matias, School of Computer Science, Tel Aviv University, Israel.
Tova Milo, School of Computer Science, Tel Aviv University, Israel.
Emmanuel Pontikakis, Department of Computer Science, Stanford University.
Wang-Chiew Tan, Univ. of California Santa Cruz.
Radek Vingralek, Google Inc.

Thesis Advisor and Postgraduate-Scholar Sponsor

Deepavali Bhagwat, Department of Computer Science, University of California Santa Cruz.
Doxa Chatzopoulou, Department of Computer Science, University of California Santa Cruz.
Jyotsna Gangwar, Department of Computer Science, University of California Santa Cruz.
Jonathan Finger, Department of Computer Science, University of California Santa Cruz.
Veronica Mayorga, Department of Computer Science, University of California Santa Cruz.
Karl Schnaitter, Department of Computer Science, University of California Santa Cruz.
Josh Spiegel, Department of Computer Science, University of California Santa Cruz.
Veronica Mayorga, University of California Santa Cruz.
Jonathan Finger, University of California Santa Cruz.
(Total number of graduate students advised: 9)